AMG for Linear Systems Obtained by Local Elimination

Tzanio Kolev

tzanio@llnl.gov

Lawrence Livermore National Laboratory

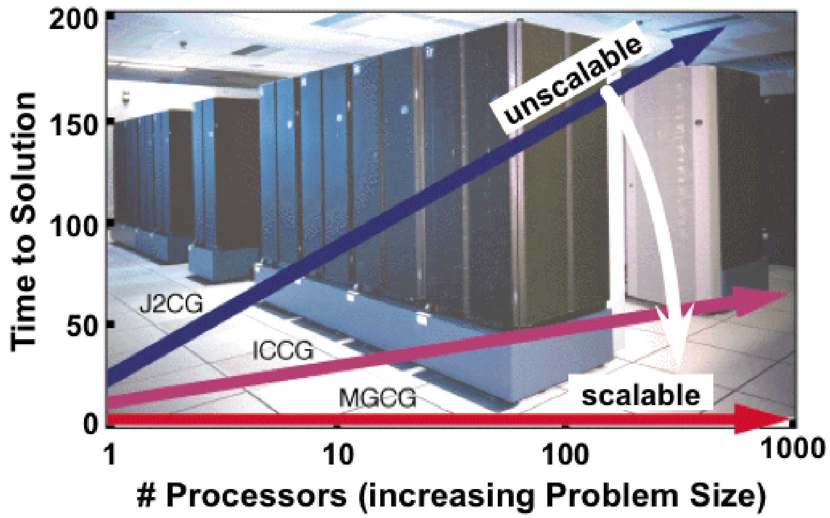
joint work with *Thomas Brunner* and *Robert Falgout*

May 4, 2010

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. LLNL-PRES-429858

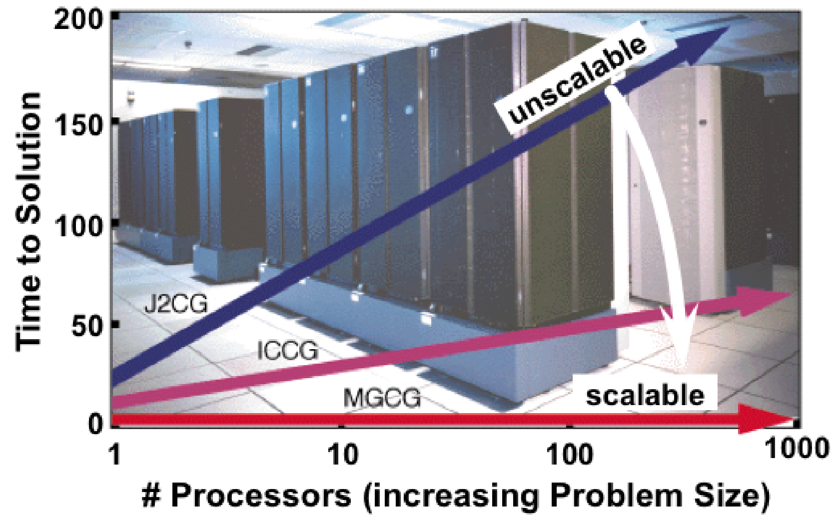
Parallel scalability and Multigrid

▷ Scalability is a central issue for large-scale parallel computing

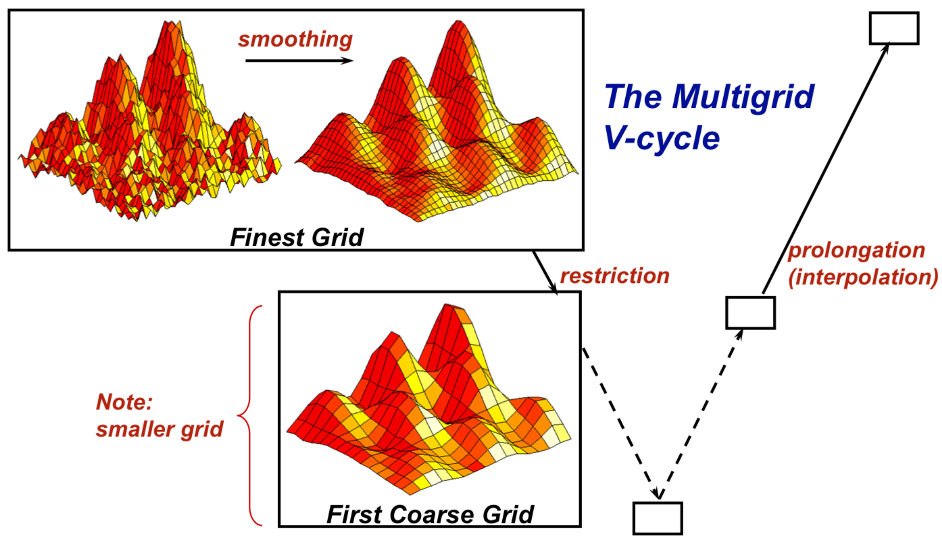


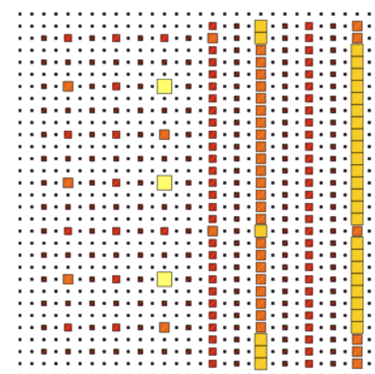
Parallel scalability and Multigrid

▷ Scalability is a central issue for large-scale parallel computing



▷ Multigrid uses coarse grids to efficiently damp out error components





AMG coarse grids

AMG for scalar diffusion

▷ Variable-coefficient Poisson

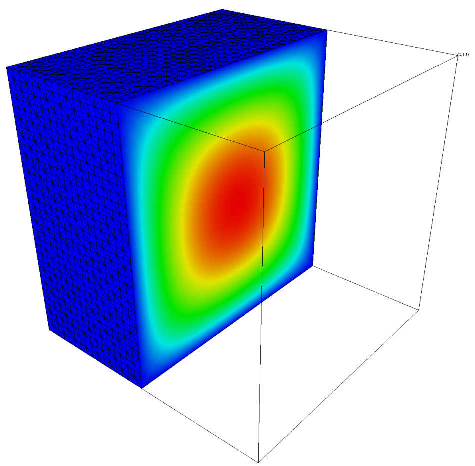
$$-\nabla \cdot \sigma \nabla u = f \quad \text{in } \Omega\,, \qquad u = 0 \quad \text{on } \partial\Omega\,.$$

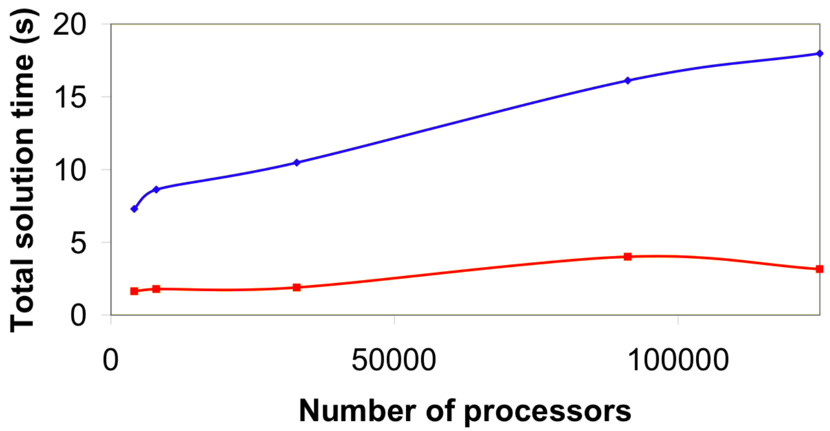
AMG for scalar diffusion

▷ Variable-coefficient Poisson

$$-\nabla \cdot \sigma \nabla u = f \quad \text{in } \Omega\,, \qquad u = 0 \quad \text{on } \partial\Omega\,.$$

▷ Weak scaling (15K/cpu) up to **128K processors** (total size 2B)



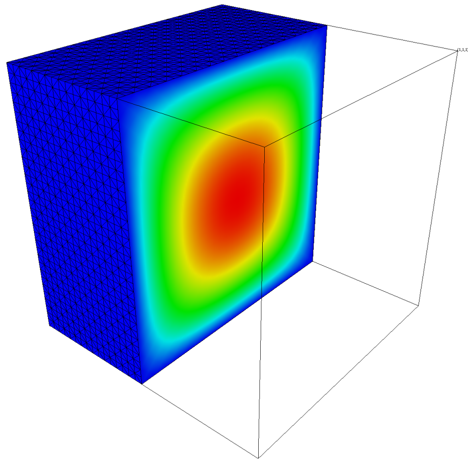


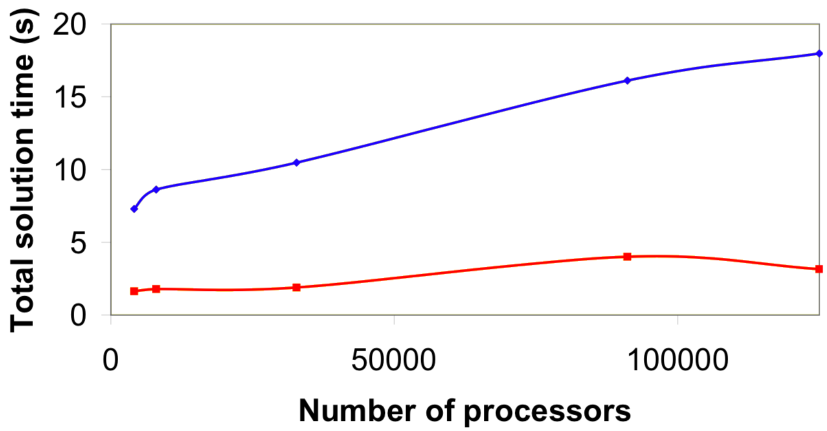
AMG for scalar diffusion

▷ Variable-coefficient Poisson

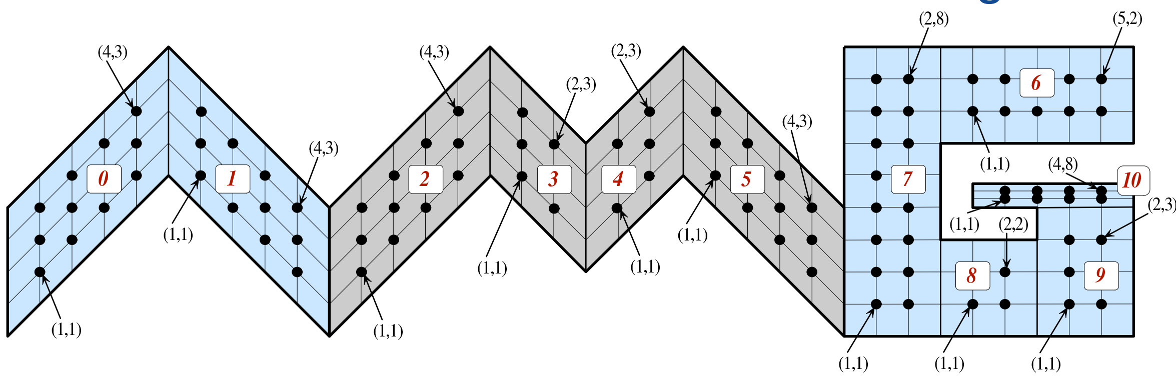
$$-\nabla \cdot \sigma \nabla u = f \quad \text{in } \Omega\,, \qquad u = 0 \quad \text{on } \partial\Omega\,.$$

▷ Weak scaling (15K/cpu) up to **128K processors** (total size 2B)





▷ Performance remains scalable on unstructured grids



26B unknowns on 98K processors took **210s** (16 iterations)

AMS for electromagnetic diffusion

▷ Second order definite Maxwell

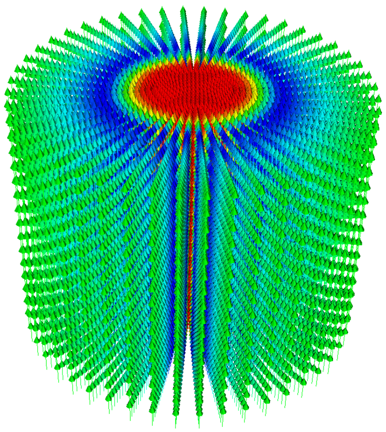
$$\nabla \times \mu^{-1} \nabla \times \boldsymbol{e} + \sigma\, \boldsymbol{e} = \boldsymbol{f}$$

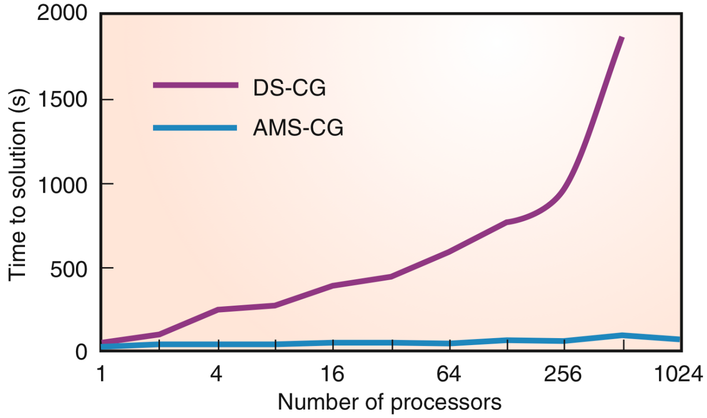
AMS for electromagnetic diffusion

▷ Second order definite Maxwell

$$\nabla \times \mu^{-1} \nabla \times \boldsymbol{e} + \sigma\, \boldsymbol{e} = \boldsymbol{f}$$

▷ Weak scaling (70K/cpu) up to **1K processors** (total size 83M)



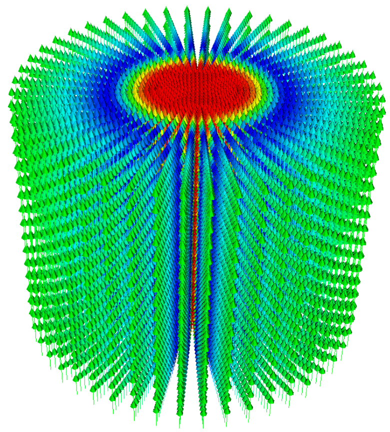


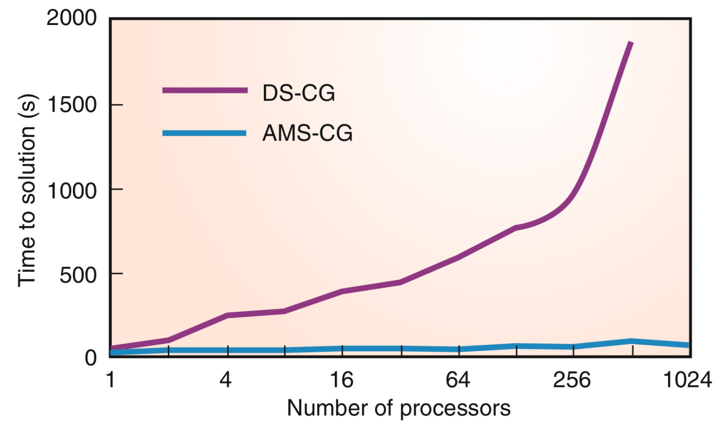
AMS for electromagnetic diffusion

▷ Second order definite Maxwell

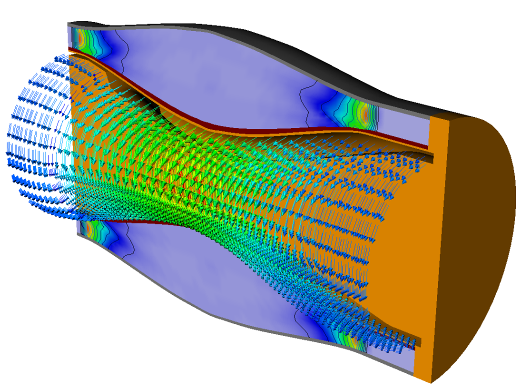
$$\nabla \times \mu^{-1} \nabla \times \boldsymbol{e} + \sigma\, \boldsymbol{e} = \boldsymbol{f}$$

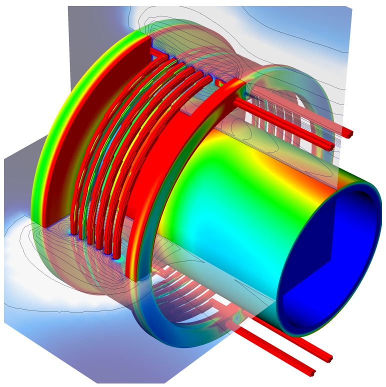
▷ Weak scaling (70K/cpu) up to **1K processors** (total size 83M)

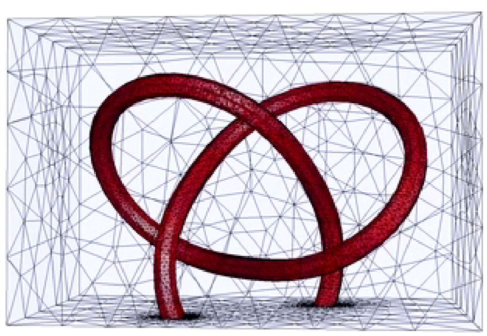




▷ Performance remains scalable on unstructured grids







1.2B unknowns on 1.9K processors took **355s** (23 iterations)

AMG research topics

- **Parallel smoothers in AMG**
  - ▷ critical component of AMG, not easy to parallelize
  - ▷ polynomial smoothers
  - ▷ hybrid Gauss-Seidel
    - convergence properties degrade, but AMG smoothing properties remain independent of number of processors (for large enough size per processor)
  - ▷ smoothing analysis based on the two-level AMG convergence theory of Falgout and Vassilevski (SINUM 2004)
- **Adaptive AMG**
  - ▷ black box (discovers the local nature of smoothness)
  - ▷ applicable to a wide range of problems (QCD)
  - ▷ theory for interpolation based on local least-squares fit of global spectrum (related to Brandt's Bootstrap AMG)
- **AMG for linear systems obtained by local elimination**

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate “interior” degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate "interior" degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

- Reduced problem

$$\boxed{\mathrm{Sx}_r = \mathrm{b}_r}$$

  - ▷ the Schur complement $\mathrm{S} = \mathrm{A}_{rr} - \mathrm{A}_{ri}\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}$ is *sparse*

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate “interior” degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

- Reduced problem

$$\boxed{\mathrm{Sx}_r = \mathrm{b}_r}$$

  - ▷ the Schur complement $\mathrm{S} = \mathrm{A}_{rr} - \mathrm{A}_{ri}\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}$ is *sparse*

- Is this a good idea?

Local elimination

- Original problem

$$\boxed{\mathrm{A}\mathrm{x} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate “interior” degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

- Reduced problem

$$\boxed{\mathrm{S}\mathrm{x}_r = \mathrm{b}_r}$$

  - ▷ the Schur complement $\mathrm{S} = \mathrm{A}_{rr} - \mathrm{A}_{ri}\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}$ is *sparse*

- Is this a good idea? – S has a smaller size, but does it require less memory?

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate "interior" degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

- Reduced problem

$$\boxed{\mathrm{S}\mathrm{x}_r = \mathrm{b}_r}$$

  - ▷ the Schur complement $\mathrm{S} = \mathrm{A}_{rr} - \mathrm{A}_{ri}\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}$ is *sparse*

- Is this a good idea? – if AMG works for A, will it also work for S?

Local elimination

- Original problem

$$\boxed{\mathrm{Ax} = \mathrm{b}}$$

  - ▷ A - FEM for scalar/electromagnetic diffusion
  - ▷ want to solve it with Algebraic Multigrid (AMG)

- Eliminate “interior” degrees of freedom

$$\mathrm{A} = \begin{pmatrix} \mathrm{A}_{ii} & \mathrm{A}_{ir} \\ \mathrm{A}_{ri} & \mathrm{A}_{rr} \end{pmatrix}$$

  - ▷ *local elimination* $\rightarrow \mathrm{A}_{ii}$ *is block-diagonal*

- Reduced problem

$$\boxed{\mathrm{Sx}_r = \mathrm{b}_r}$$

  - ▷ the Schur complement $\mathrm{S} = \mathrm{A}_{rr} - \mathrm{A}_{ri}\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}$ is *sparse*

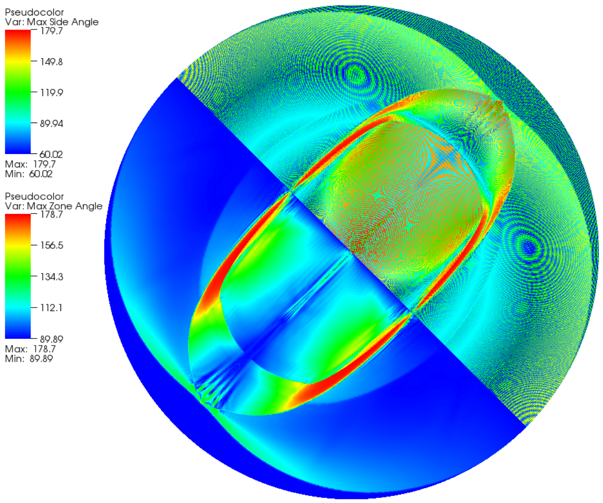
- Is this a good idea? – can we solve larger problems faster?

Motivating application

- Large-scale parallel multi-physics simulation code
  - ▷ Electromagnetic diffusion model
  - ▷ Second order definite Maxwell

$$\nabla \times \frac{\Delta t}{\mu} \nabla \times \boldsymbol{e} + \sigma\, \boldsymbol{e} = \boldsymbol{f}$$

  - ▷ Lowest order edge elements
  - ▷ Large jumps in σ
  - ▷ Support for pure void zones

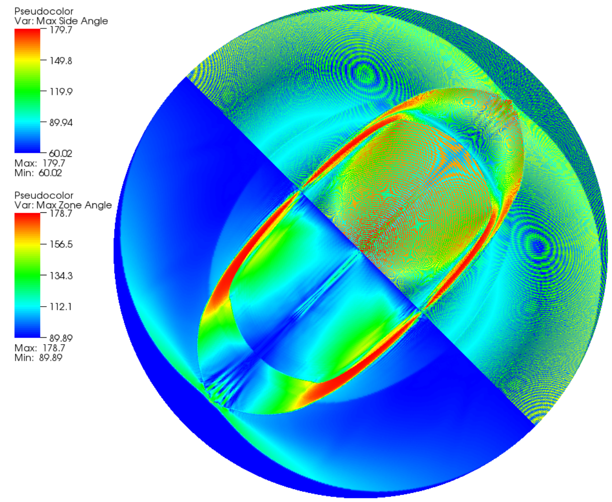


Motivating application

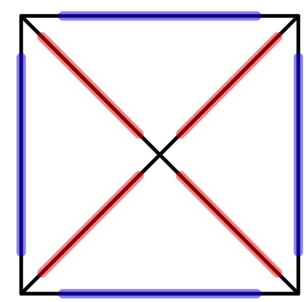
- Large-scale parallel multi-physics simulation code
  - ▷ Electromagnetic diffusion model
  - ▷ Second order definite Maxwell

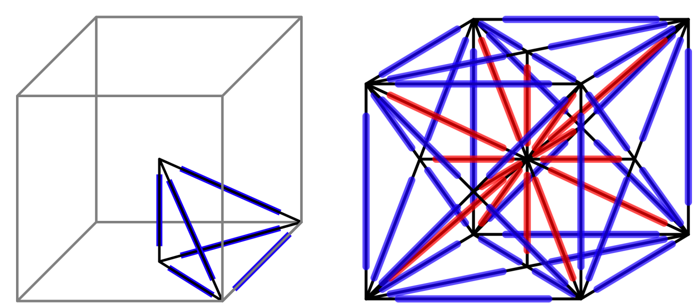
$$\nabla \times \frac{\Delta t}{\mu} \nabla \times \boldsymbol{e} + \sigma \, \boldsymbol{e} = \boldsymbol{f}$$

  - ▷ Lowest order edge elements
  - ▷ Large jumps in σ
  - ▷ Support for pure void zones



- Sub-zonal discretization:
  - ▷ Initial quad/hex mesh split into 4/24 tri/tet elements
  - ▷ XY, RZ and $3D$ models lead to
    - 2D Poisson
    - 2D Maxwell
    - 3D Maxwell

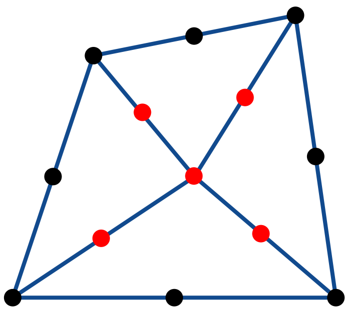




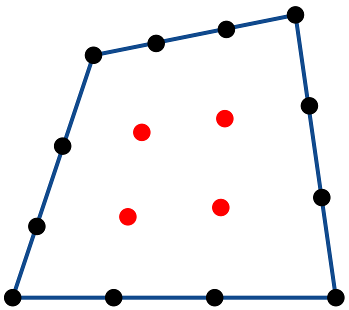
Memory – the case of no fill-in

- Static condensation
  - ▷ introduced by E. Wilson in 1974 to

    "eliminate the internal degrees of freedom in a quadrilateral finite element constructed from four triangles"



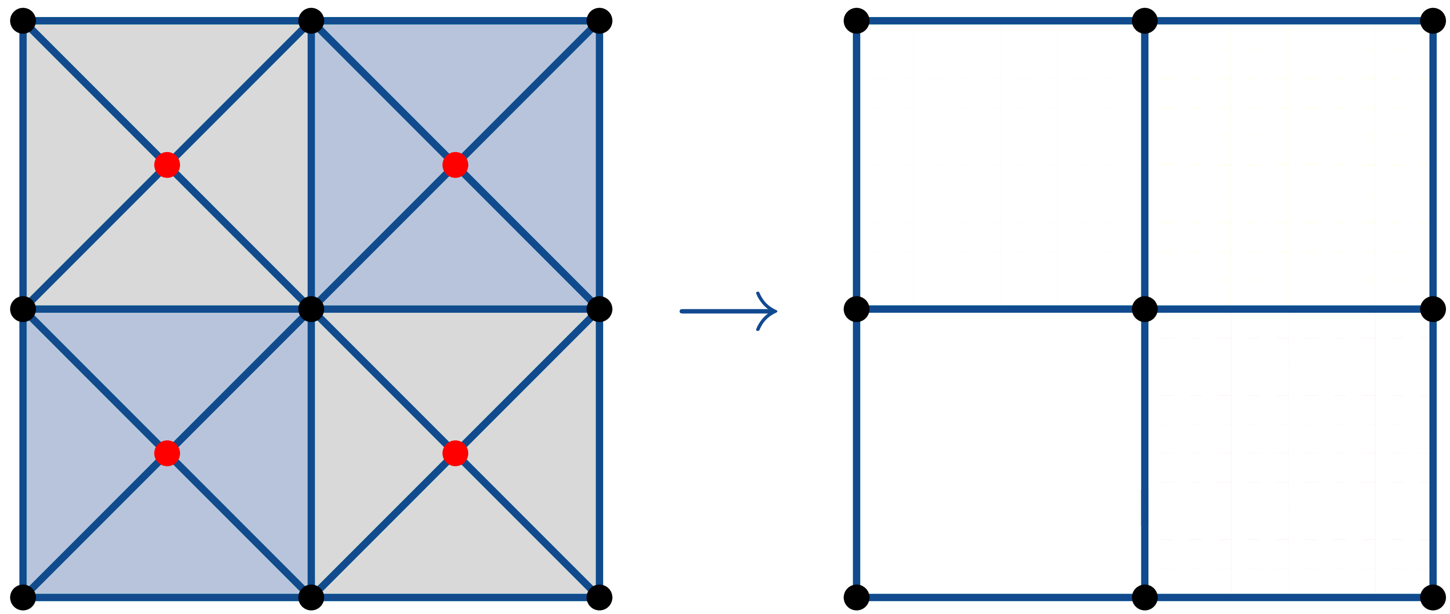
  - ▷ frequently used to eliminate the interior degrees of freedom in high-order FEM



  - ▷ *sparsity of* A_{rr} *is not increased!*

An element reduction approach

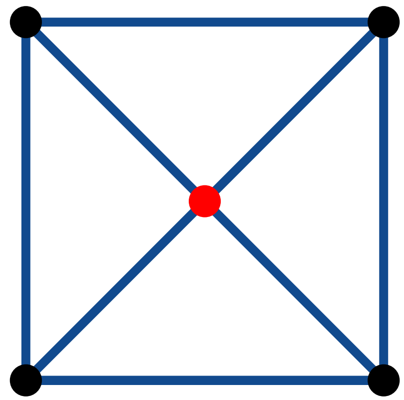
1. Choose the set of reduced elements
2. Determine the interior dofs
3. Connect reduced unknowns



- Reduced FEM discretization
  - ▷ Reduced elements
  - ▷ Reduced degrees of freedom
  - ▷ Reduced element matrices (local Schur complements)

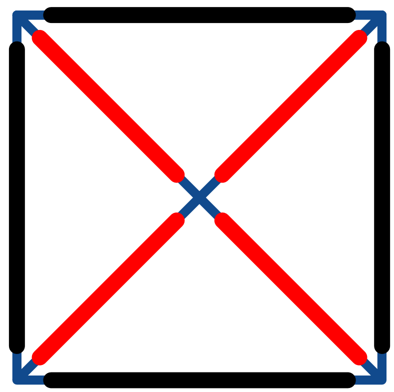
Element reduction in 2D

- The XY case (2D nodal FEM)



  - ▷ Asymptotically $nrows(\mathrm{A})/nrows(\mathrm{S}) \sim 2$, $nnz(\mathrm{A})/nnz(\mathrm{S}) \sim 1.6$

- The RZ case (2D edge FEM)

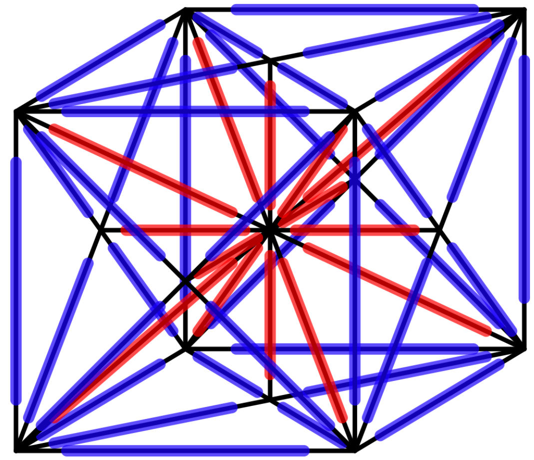


  - ▷ Asymptotically $nrows(\mathrm{A})/nrows(\mathrm{S}) \sim 3$, $nnz(\mathrm{A})/nnz(\mathrm{S}) \sim 2.1$

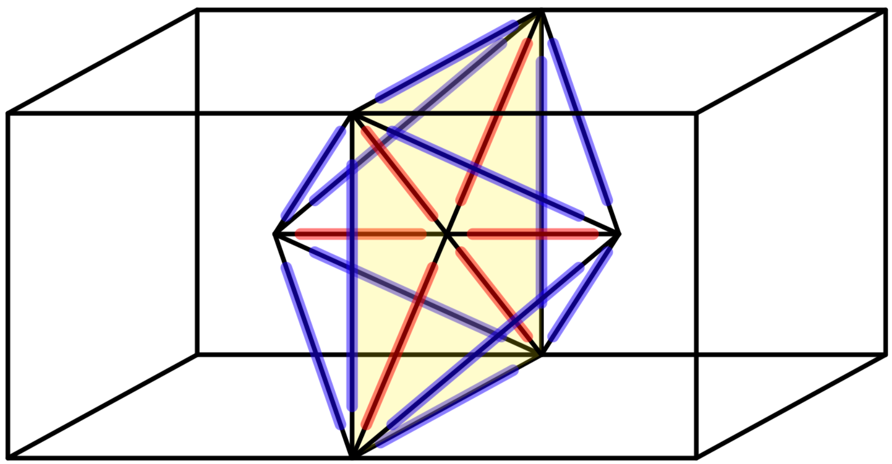
- In both cases we recover the associated quad mesh, *but not the quad-based discretization!*

Element reduction in 3D

- The full $3D$ case (3D edge FEM)
  - ▷ S_H : hexahedral reduced elements, $nnz(\mathrm{A})/nnz(\mathrm{S}) \sim 0.4$



  - ▷ S_O : octahedral reduced elements, $nnz(\mathrm{A})/nnz(\mathrm{S}) \sim 1.4$



- Octahedral reduction is the best in terms of memory usage!

If AMG works for A, will it also work for S?

Schur complement properties

- The Schur complement inherits a lot of solver-friendly properties from the original matrix
  - ▷ S can be assembled locally
  - ▷ If A is an M-matrix, so is S
  - ▷ $\kappa(\mathrm{S}) \le \kappa(\mathrm{A})$
- S can be seen as a coarse-grid matrix corresponding to interpolation by a harmonic extension

$$\mathrm{S} = \mathrm{P}^t \mathrm{A} \mathrm{P}\,, \quad \text{where} \quad \mathrm{P} = \begin{pmatrix} -\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir} \\ \mathrm{I} \end{pmatrix}.$$

- Energy minimization property

$$(\mathrm{S}\mathrm{x}_r, \mathrm{x}_r) = \inf_{\mathrm{x}|_r = \mathrm{x}_r} (\mathrm{A}\mathrm{x}, \mathrm{x})$$

- In particular, $\mathrm{D}_\mathrm{S} \le \mathrm{D}_\mathrm{A}$, where $\mathrm{D}_\mathrm{M} := \mathrm{diag}(\mathrm{M})$.

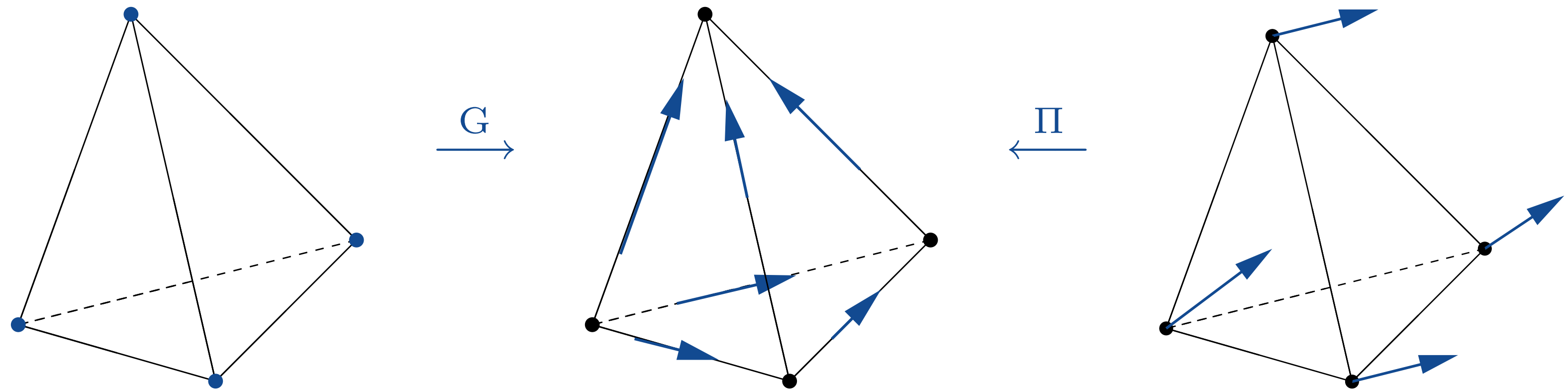
AMG solvers

- Require knowledge of near nullspace: $\mathrm{Ae} \approx 0$
- Classical AMG for Poisson problems
  - ▷ near nullspace is locally constant
  - ▷ coarsening and interpolation based on strength of connection: e_i strongly depends on e_j if

$$-\mathrm{A}_{ij} \geq \theta \max_{k \neq i} \{-\mathrm{A}_{ik}\}$$

  - ▷ $0 < \theta \leq 1$ is the strength threshold parameter.
- Auxiliary-space Maxwell Solver (AMS) for definite Maxwell
  - ▷ near nullspace is large, includes local gradients
  - ▷ based on the finite element HX decomposition by Hiptmair and Xu
  - ▷ two (auxiliary space) V-cycles requiring discrete gradient and Nedelec interpolation matrices

AMS interpolation operators



- Discrete gradient matrix G corresponds to the mapping

$$\varphi \in S_h \mapsto \nabla\varphi \in \boldsymbol{V}_h \,,$$

G describes the edges of the mesh in terms of its vertices.

- The Nedelec interpolation operator $\boldsymbol{\Pi}_h$ transfers linear vector fields $\boldsymbol{\varphi} \in \boldsymbol{S}_h \equiv S_h^3$ into $\boldsymbol{V}_h$:

$$\boldsymbol{\Pi}_h \boldsymbol{\varphi} = \sum_e \left(\int_e \boldsymbol{\varphi} \cdot \boldsymbol{t}_e \, ds \right) \boldsymbol{\Phi}_e \,.$$

$\Pi = [\Pi_x \; \Pi_y \; \Pi_z]$ – *the matrix representation of* $\boldsymbol{\Pi}_h$ *can be computed based on* G *and the coordinates of the vertices*.

HX decomposition and AMS

- Hiptmair-Xu decomposition: any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ can be split into

$$\boxed{\boldsymbol{u}_h = \boldsymbol{v}_h + \nabla p_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h}$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $p_h \in S_h$ and $\boldsymbol{z}_h \in \boldsymbol{S}_h$ satisfy

$$h^{-1}\|\boldsymbol{v}_h\|_{\mathbf{0}} + \|\boldsymbol{z}_h\|_{\mathbf{1}} \leq C\,\|\nabla \times \boldsymbol{u}_h\|_{\mathbf{0}}\,, \qquad \|\nabla p_h\|_{\mathbf{0}} \leq C\,\|\boldsymbol{u}_h\|_{\mathbf{0}}\,.$$

R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in $\boldsymbol{H}(\mathbf{curl})$ and $\boldsymbol{H}(\mathrm{div})$ spaces, *SINUM*, *2007*.

HX decomposition and AMS

- Hiptmair-Xu decomposition: any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ can be split into

$$\boxed{\boldsymbol{u}_h = \boldsymbol{v}_h + \nabla p_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h}$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $p_h \in S_h$ and $\boldsymbol{z}_h \in \boldsymbol{S}_h$ satisfy

$$h^{-1}\|\boldsymbol{v}_h\|_{\mathbf{0}} + \|\boldsymbol{z}_h\|_{\mathbf{1}} \le C\,\|\nabla \times \boldsymbol{u}_h\|_{\mathbf{0}}\,, \qquad \|\nabla p_h\|_{\mathbf{0}} \le C\,\|\boldsymbol{u}_h\|_{\mathbf{0}}\,.$$

R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in $\boldsymbol{H}(\mathbf{curl})$ and $\boldsymbol{H}(\mathrm{div})$ spaces, *SINUM*, *2007*.

- AMS implementation

$$\boxed{\mathrm{B} = \mathrm{R} + \mathrm{G} \mathrm{B} \mathrm{G}^T + \Pi\, \mathrm{B}_v\, \Pi^T}$$

where

- R is a point smoother for A.
- B is an AMG V-cycle for $\mathrm{G}^{\mathrm{T}}\mathrm{A}\mathrm{G}$.
- B_v is an AMG V-cycle for $\Pi^{\mathrm{T}}\mathrm{A}\Pi$ $\quad$ ($\|\boldsymbol{\Pi}_h \boldsymbol{z}_h\|_{\boldsymbol{H}(\mathbf{curl})} \lesssim \|\boldsymbol{z}_h\|_{\mathbf{1}}$).

Near-nullspace reduction

- The near nullspace of S is the restriction of the near-nullspace of A to the reduced degrees of freedom:
  - ▷ Suppose $\mathrm{Ae} \approx 0$, then $\mathrm{A}_{ii}\mathrm{e}_i + \mathrm{A}_{ir}\mathrm{e}_r \approx 0$ implies $\mathrm{e} \approx \mathrm{Pe}_r$, so

$$\mathrm{Se}_r = \mathrm{P}^t\mathrm{APe}_r \approx \mathrm{P}^t\mathrm{Ae} \approx 0\,.$$

  - ▷ On the other hand, $(\mathrm{Se}_r, \mathrm{e}_r) \approx 0$ implies $\mathrm{Ae} \approx 0$ for $\mathrm{e} = \mathrm{Pe}_r$.
  - ▷ for XY we can apply AMG directly to S

Near-nullspace reduction

- The near nullspace of S is the restriction of the near-nullspace of A to the reduced degrees of freedom:
  - ▷ Suppose $\mathrm{Ae} \approx 0$, then $\mathrm{A}_{ii}\mathrm{e}_i + \mathrm{A}_{ir}\mathrm{e}_r \approx 0$ implies $\mathrm{e} \approx \mathrm{Pe}_r$, so

$$\mathrm{Se}_r = \mathrm{P}^t\mathrm{APe}_r \approx \mathrm{P}^t\mathrm{Ae} \approx 0\,.$$

  - ▷ On the other hand, $(\mathrm{Se}_r, \mathrm{e}_r) \approx 0$ implies $\mathrm{Ae} \approx 0$ for $\mathrm{e} = \mathrm{Pe}_r$.
  - ▷ for XY we can apply AMG directly to S

- Reduced discrete gradient and nodal interpolation matrices.
  - ▷ Edge reduction implies node reduction
  - ▷ Note that the discrete gradient matrix can be partitioned as

$$\mathrm{G} = \begin{pmatrix} \mathrm{G}_{ii} & \mathrm{G}_{ir} \\ 0 & \mathrm{G}_{rr} \end{pmatrix}$$

  - ▷ The restriction of $\mathrm{Ran}(\mathrm{G})$ to reduced unknowns is $\mathrm{Ran}(\mathrm{G}_{rr})$ – the discrete gradient defined on the reduced mesh.
  - ▷ Same holds for Π, so we can apply AMS directly to S

Reduced HX decomposition

In matrix terms, the HX decomposition states that

$$\mathrm{u} = \mathrm{v} + \mathrm{Gp} + \Pi\mathrm{z}$$

such that

$$(\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{AGp}, \mathrm{Gp}) + (\mathrm{A}\Pi\mathrm{z}, \Pi\mathrm{z}) + (\mathrm{D_A v}, \mathrm{v})$$

Reduced HX decomposition

In matrix terms, the HX decomposition states that

$$\mathrm{u} = \mathrm{v} + \mathrm{G}\mathrm{p} + \Pi\mathrm{z}$$

such that

$$(\mathrm{A}\mathrm{u}, \mathrm{u}) \gtrsim (\mathrm{A}\mathrm{G}\mathrm{p}, \mathrm{G}\mathrm{p}) + (\mathrm{A}\Pi\mathrm{z}, \Pi\mathrm{z}) + (\mathrm{D}_{\mathrm{A}}\mathrm{v}, \mathrm{v})$$

Fix u_r and consider $\mathrm{u} = \mathrm{P}\mathrm{u}_r$ above. Then

$$\mathrm{u}_r = \mathrm{v}_r + \mathrm{G}_{rr}\mathrm{p}_r + \Pi_{rr}\mathrm{z}_r$$

Reduced HX decomposition

In matrix terms, the HX decomposition states that

$$\mathrm{u} = \mathrm{v} + \mathrm{G}\mathrm{p} + \Pi\mathrm{z}$$

such that

$$(\mathrm{Au},\mathrm{u}) \gtrsim (\mathrm{AGp},\mathrm{Gp}) + (\mathrm{A}\Pi\mathrm{z},\Pi\mathrm{z}) + (\mathrm{D_A v},\mathrm{v})$$

Fix u_r and consider $\mathrm{u} = \mathrm{P}\mathrm{u}_r$ above. Then

$$\mathrm{u}_r = \mathrm{v}_r + \mathrm{G}_{rr}\mathrm{p}_r + \Pi_{rr}\mathrm{z}_r$$

Therefore,

$$(\mathrm{S}\mathrm{u}_r,\mathrm{u}_r) = (\mathrm{Au},\mathrm{u}) \gtrsim (\mathrm{AGp},\mathrm{Gp}) \geq (\mathrm{S}\mathrm{G}_{rr}\mathrm{p}_r,\mathrm{G}_{rr}\mathrm{p}_r)$$

Reduced HX decomposition

In matrix terms, the HX decomposition states that

$$\mathrm{u} = \mathrm{v} + \mathrm{Gp} + \Pi \mathrm{z}$$

such that

$$(\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{AGp}, \mathrm{Gp}) + (\mathrm{A}\Pi \mathrm{z}, \Pi \mathrm{z}) + (\mathrm{D_A v}, \mathrm{v})$$

Fix u_r and consider $\mathrm{u} = \mathrm{Pu}_r$ above. Then

$$\mathrm{u}_r = \mathrm{v}_r + \mathrm{G}_{rr}\mathrm{p}_r + \Pi_{rr}\mathrm{z}_r$$

Therefore,

$$(\mathrm{Su}_r, \mathrm{u}_r) = (\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{AGp}, \mathrm{Gp}) \geq (\mathrm{SG}_{rr}\mathrm{p}_r, \mathrm{G}_{rr}\mathrm{p}_r)$$

Similarly $(\mathrm{Su}_r, \mathrm{u}_r) \gtrsim (\mathrm{S}\Pi_{rr}\mathrm{p}_r, \Pi_{rr}\mathrm{p}_r)$. Note that Π_{rr} can still be computed from G_{rr} and the coordinates of the reduced vertices.

Reduced HX decomposition

In matrix terms, the HX decomposition states that

$$\mathrm{u} = \mathrm{v} + \mathrm{G}\mathrm{p} + \Pi \mathrm{z}$$

such that

$$(\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{AGp}, \mathrm{Gp}) + (\mathrm{A}\Pi\mathrm{z}, \Pi\mathrm{z}) + (\mathrm{D_A v}, \mathrm{v})$$

Fix u_r and consider $\mathrm{u} = \mathrm{P}\mathrm{u}_r$ above. Then

$$\mathrm{u}_r = \mathrm{v}_r + \mathrm{G}_{rr}\mathrm{p}_r + \Pi_{rr}\mathrm{z}_r$$

Therefore,

$$(\mathrm{S}\mathrm{u}_r, \mathrm{u}_r) = (\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{AGp}, \mathrm{Gp}) \geq (\mathrm{S}\mathrm{G}_{rr}\mathrm{p}_r, \mathrm{G}_{rr}\mathrm{p}_r)$$

Similarly $(\mathrm{S}\mathrm{u}_r, \mathrm{u}_r) \gtrsim (\mathrm{S}\Pi_{rr}\mathrm{p}_r, \Pi_{rr}\mathrm{p}_r)$. Note that Π_{rr} can still be computed from G_{rr} and the coordinates of the reduced vertices.

Finally,

$$(\mathrm{S}\mathrm{u}_r, \mathrm{u}_r) = (\mathrm{Au}, \mathrm{u}) \gtrsim (\mathrm{D_A v}, \mathrm{v}) \geq (\mathrm{D_S}\mathrm{v}_r, \mathrm{v}_r)$$

Reduced subspace problems

- $\mathrm{G}_{rr}^T\mathrm{S}\mathrm{G}_{rr}$ is the Schur complement of $\mathrm{G}^T\mathrm{A}\mathrm{G}$
  - ▷ classical AMG works for the reduced subspace problems
- Commuting diagram

$$\boxed{\mathrm{P}\mathrm{G}_{rr} = \mathrm{G}\mathrm{P}_n}$$

where P_n – nodal $\mathrm{G}^T\mathrm{A}\mathrm{G}$-harmonic extension:

$$\mathrm{P}_n = \begin{pmatrix} -(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ii}^{-1}(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ir} \\ \mathrm{I} \end{pmatrix}$$

- Proof

$$lhs_i = -\mathrm{A}_{ii}^{-1}\mathrm{A}_{ir}\mathrm{G}_{rr}\,, \quad rhs_i = -\mathrm{G}_{ii}(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ii}^{-1}(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ir} + \mathrm{G}_{ir}$$

Note that

$$(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ii} = \mathrm{G}_{ii}^T\mathrm{A}_{ii}\mathrm{G}_{ii}\,, \quad (\mathrm{G}^T\mathrm{A}\mathrm{G})_{ir} = \mathrm{G}_{ii}^T\mathrm{A}_{ii}\mathrm{G}_{ir} + \mathrm{G}_{ii}^T\mathrm{A}_{ir}\mathrm{G}_{rr}$$

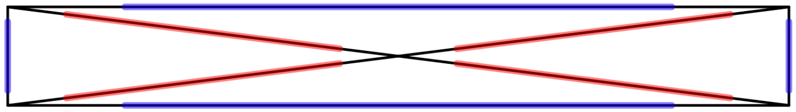
Thus

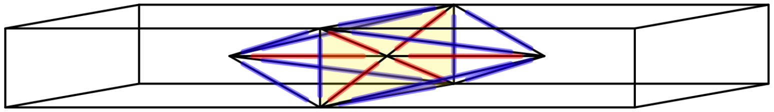
$$(\mathrm{G}^T\mathrm{A}\mathrm{G})_{ii}\mathrm{G}_{ii}^{-1}rhs_i = \mathrm{G}_{ii}^T\mathrm{A}_{ii}rhs_i = -\mathrm{G}_{ii}^T\mathrm{A}_{ir}\mathrm{G}_{rr} = \mathrm{G}_{ii}^T\mathrm{A}_{ii}lhs_i$$

- Now

$$\mathrm{P}_n^T\mathrm{G}^T\mathrm{A}\mathrm{G}\mathrm{P}_n = \mathrm{G}_{rr}^T\mathrm{P}^T\mathrm{A}\mathrm{P}\mathrm{G}_{rr} = \mathrm{G}_{rr}^T\mathrm{S}\mathrm{G}_{rr}$$

Meshes with stretched elements





- A common occurrence in the motivating applications
- In 2D the reduction process will eliminate badly shaped triangles. In 3D the improvement is only marginal.
- Compare reduced stencil with the standard Q_1 FEM stencil (where AMG does not work with $\theta = 0.25$).

-1	-6	-1
2	12	2
-1	-6	-1

-1	-4	-1
2	8	2
-1	-4	-1

  ▷ Introducing and then eliminating the (artificial) interior unknowns leads to a better discretization for Multigrid!

- We expect improved performance on stretched grids in 2D.

Can we solve larger problems faster?

AMG solvers used

- Tests with *BoomerAMG* and *AMS* from hypre

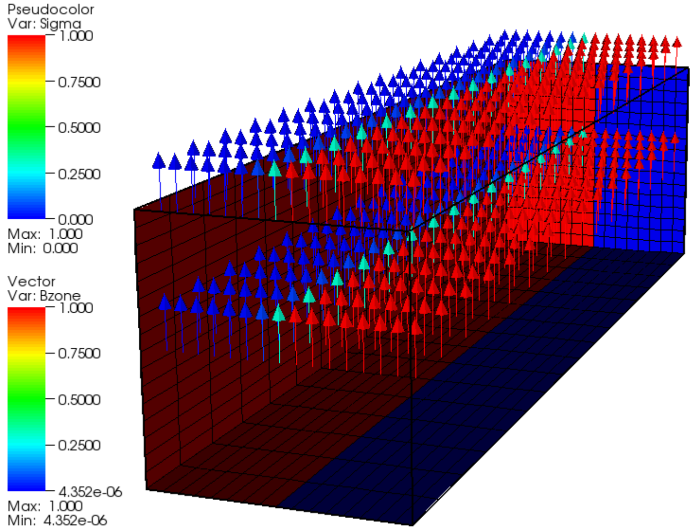
```
HYPRE_Solver solver;
HYPRE_AMSCreate(&solver);

/* Set discrete gradient matrix */
HYPRE_AMSSetDiscreteGradient(solver, G);
/* Set vertex coordinates */
HYPRE_AMSSetCoordinateVectors(solver, X, Y, Z);

HYPRE_AMSSetup(solver, A, b, x);
HYPRE_AMSSolve(solver, A, b, x);
```

- Both applied as preconditioners in CG for the reduced problem.
- Using BoomerAMG's low-complexity coarsening and long-range interpolation options.
- Using the zero-conductivity version of AMS for problems with pure void.
- Notation: θ, σ_{nc}/σ_c, ε, n_{it}, t_{setup}, t_{solve}, t.

Box problem



- Magnetic field diffuses through void and into material.
- Simplified serial test to vary conductivity ratio, aspect ratio and solver parameters.

problem	N	nnz
XY	33,025 / 16,641	230,145 / 148,225 ($\times$1.6)
RZ	98,560 / 33,024	491,776 / 229,632 ($\times$2.1)
$3D$	239,260 / 90,460	3,724,060 / 2,658,460 ($\times$1.4)

- $\Delta t/\mu \sim 10^{-3}$
- AMS-CG convergence tolerance 10^{-10}.

Box problem - XY

- Comparison of overall solution times
- $\theta = 0.34$, $\sigma_{nc}/\sigma_c = 0$

$1/\varepsilon$	n_{it}	$t_{assemble}$	t_{solver}	t
1	7/ 7	0.33/0.24	0.34/0.13	$\times$1.8
2	13/ 8	0.30/0.21	0.43/0.13	$\times$2.2
4	12/ 8	0.28/0.21	0.38/0.13	$\times$2.0
8	12/12	0.28/0.21	0.37/0.17	$\times$1.7
16	16/12	0.28/0.22	0.45/0.19	$\times$1.7
32	24/11	0.28/0.21	0.62/0.17	$\times$2.4
64	35/ 9	0.29/0.21	0.90/0.15	$\times$3.2
128	40/ 7	0.32/0.22	1.10/0.14	$\times$4.0
256	45/ 7	0.30/0.24	1.24/0.16	$\times$3.8
512	45/ 7	0.28/0.23	1.12/0.13	$\times$3.9
1024	46/ 7	0.32/0.24	1.32/0.16	$\times$4.0
2048	46/ 7	0.29/0.24	1.29/0.16	$\times$3.9
4096	46/ 7	0.30/0.25	1.37/0.15	$\times$4.2

- Note the reduced setup time and that when we have the same number of iterations ($\varepsilon = 1$) there is still a factor of 1.8 speedup.

Box problem - XY

- Reduced problem – dependence on σ
- $\theta = 0.4$

	σ_{nc}/σ_c					
$1/\varepsilon$	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	0
1	7	7	7	7	7	7
2	7	8	7	7	7	7
4	7	8	8	8	8	8
8	7	8	8	8	8	8
16	7	8	8	8	8	8
32	7	8	8	8	8	8
64	6	7	7	7	7	7
128	6	6	6	6	6	6
256	7	6	6	6	6	6
512	8	7	7	7	7	7
1024	8	7	7	7	7	7
2048	8	7	7	7	7	7
4096	8	7	7	7	7	7

- Number of iterations independent σ_{nc}/σ_c and ε!

Box problem - RZ

- Comparison of overall solution times – pure void
- $\theta = 0.17$, $\sigma_{nc}/\sigma_c = 0$

$1/\varepsilon$	n_{it}	$t_{assemble}$	t_{solver}	t
1	8/ 8	1.74/0.75	11.9/3.96	$\times$2.9
2	8/ 8	1.66/0.71	12.2/4.08	$\times$2.9
4	10/ 8	1.76/0.72	14.2/4.09	$\times$3.3
8	16/ 8	1.73/0.69	20.1/3.84	$\times$4.8
16	28/ 9	1.65/0.71	26.7/4.01	$\times$6.0
32	45/11	1.35/0.71	34.6/4.69	$\times$6.7
64	74/14	1.23/0.71	50.1/5.05	$\times$8.9
128	125/18	1.27/0.71	80.9/6.53	$\times$11.
256	211/24	1.26/0.71	138./8.31	$\times$15.
512	362/28	1.49/0.69	236./9.47	$\times$23.
1024	500/30	1.25/0.71	315./9.70	$\times$30.
2048	707/31	1.26/0.71	352./10.4	$\times$32.
4096	828/33	1.04/0.68	407./11.0	$\times$35.

- This is AMG for the Schur complement of a singular matrix!
- Iteration times increase, but we need less of them for small ε.

Box problem - RZ

- Reduced problem – dependence on σ
- $\theta = 0.17$

	σ_{nc}/σ_c					
$1/\varepsilon$	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	0
1	8	8	8	8	8	8
2	8	8	8	8	8	8
4	8	8	8	8	8	8
8	8	9	9	8	8	8
16	8	9	9	9	9	9
32	11	11	11	11	11	11
64	14	14	14	14	14	14
128	19	18	18	18	18	18
256	27	23	23	23	23	24
512	32	28	28	28	28	28
1024	56	47	47	47	47	30
2048	65	52	53	53	53	31
4096	71	58	57	57	57	33

- Not sensitive to jumps in σ; improved robustness for $\sigma_{nc} = 0$.

Box problem - $3D$

- Comparison of overall solution times
- $\theta = 0.5$, $\sigma_{nc}/\sigma_c = 10^{-4}$

$1/\varepsilon$	n_{it}	$t_{assemble}$	t_{solver}	t
1	9/ 8	6.58/5.04	40.3/17.3	$\times$2.1
2	9/ 8	7.34/5.14	47.6/16.1	$\times$2.6
4	16/ 9	7.10/5.07	67.6/16.5	$\times$3.5
8	29/ 15	7.71/5.15	111./23.8	$\times$4.1
16	49/ 26	7.40/5.15	178./37.1	$\times$4.4
32	79/ 42	8.15/5.11	262./55.1	$\times$4.5
64	121/ 66	7.83/4.95	372./85.1	$\times$4.2
128	180/107	6.66/5.23	546./138.	$\times$3.8
256	248/163	7.73/5.23	807./205.	$\times$3.9
512	332/234	8.65/5.01	1025/278.	$\times$3.7
1024	485/297	7.73/4.27	1327/299.	$\times$4.4
2048	677/268	6.58/3.65	1968/213.	$\times$9.1
4096	1064/250	7.55/4.19	3862/256.	$\times$15.

- Convergence deteriorates significantly on stretched grids.
- Performance is practically uniform in θ.

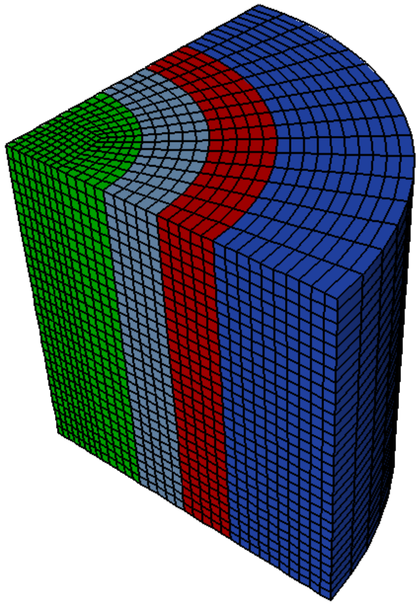
Box problem - $3D$

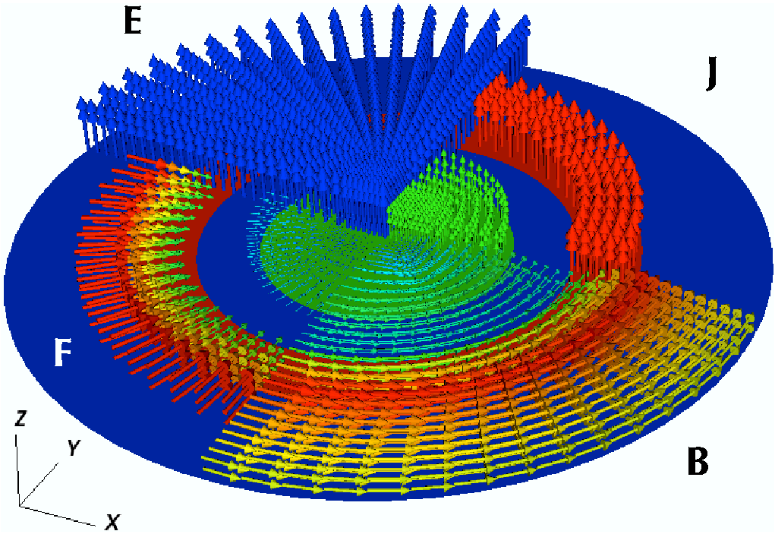
- Reduced problem – dependence on σ
- $\theta = 0.5$

	σ_{nc}/σ_c					
$1/\varepsilon$	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	0
1	8	8	8	8	8	8
2	8	8	8	8	8	8
4	9	9	9	9	9	9
8	15	15	15	15	15	15
16	26	26	26	26	26	26
32	41	42	42	42	42	42

- Convergence is independent of jumps in σ

Coax problem





- Four coaxial cylindrical conductors with varying conductivity.
- Mock up for the kinds of jumps in Z-pinch simulations.
- $\sigma \sim \{10^{-2}, 10^{-8}, 10^{-2}, 0\}$, $\Delta t/\mu \sim 10^{-4}$.
- XY and RZ cases correspond to the top and front sides.
- $\theta = 0.5$, $\varepsilon = 1$
- AMS-CG convergence tolerance 10^{-10}.

Coax problem - XY

np	N	n_{it}
1	15,013 / 7,589	13/10
4	59,721 / 30,025	14/10
16	238,225 / 119,441	15/13
64	951,585 / 476,449	17/15
256	3,803,713 / 1,903,169	20/17

- $\mathbb{S}$ has 1.6 times fewer nonzero entries compared to $\mathbb{A}$.

np	$t_{assemble}$	t_{setup}	t_{solve}	t
1	0.13/0.10	0.07/0.03	0.21/0.08	$\times$1.9
4	0.14/0.11	0.09/0.05	0.23/0.09	$\times$1.8
16	0.17/0.13	0.12/0.07	0.47/0.19	$\times$1.9
64	0.26/0.14	0.27/0.19	0.75/0.36	$\times$1.8
256	0.22/0.17	0.98/0.75	1.58/0.79	$\times$1.6

Coax problem - RZ

np	N	n_{it}
1	21,720 / 7,320	10/11
4	86,640 / 29,040	10/12
16	346,080 / 115,680	11/13
64	1,383,360 / 461,760	12/13

- $\mathbb{S}$ has 2.1 times fewer nonzero entries compared to $\mathbb{A}$.

np	$t_{assemble}$	t_{setup}	t_{solve}	t
1	0.19/0.12	0.21/0.09	0.56/0.27	$\times$2.0
4	0.19/0.11	0.34/0.15	0.84/0.43	$\times$2.0
16	0.24/0.13	0.54/0.26	1.70/0.72	$\times$2.2
64	0.24/0.14	1.23/0.63	2.48/1.10	$\times$2.1

Coax problem - $3D$

np	N	n_{it}
1	208,370 / 78,774	12/10
8	1,640,728 / 621,224	13/10
64	1,3021,568 / 4,934,592	14/11
512	103,756,864 / 39,337,280	15/14

- $\mathbb{S}$ has 1.4 times fewer nonzero entries compared to $\mathbb{A}$.

np	$t_{assemble}$	t_{setup}	t_{solve}	t
1	3.58/2.57	10.0/3.17	23.8/6.91	$\times$2.9
8	4.03/2.77	32.5/6.95	55.1/10.9	$\times$4.4
64	4.60/3.15	80.3/18.5	113./28.8	$\times$3.9
512	6.47/3.32	174./75.5	210./113.	$\times$2.0

Conclusions

- The AMG/AMS solvers perform well in practice when applied to reduced scalar/electromagnetic diffusion problems.
- Typical speed-up factors in the considered simulations were **1.6-4.2** (XY), **2.0-36** (RZ) and **2.0-4.5** ($3D$).
- Typical memory reduction: **1.6** (XY), **2.1** (RZ) and **1.4** ($3D$).
- Reduced HX: AMS works on Schur complements!

$$\boxed{(\mathrm{Su}_r, \mathrm{u}_r) \gtrsim (\mathrm{SG}_{rr}\mathrm{p}_r, \mathrm{G}_{rr}\mathrm{p}_r) + (\mathrm{S}\Pi_{rr}\mathrm{z}_r, \Pi_{rr}\mathrm{z}_r) + (\mathrm{D_S}\mathrm{v}_r, \mathrm{v}_r)}$$

- The elimination process leads to lower assembly, solver/setup times and faster iterations, independent of jumps in σ.
- Reduction can be easily modified to handle the pure void case.
- Some details can be found in

R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in $\boldsymbol{H}(\mathbf{curl})$ and $\boldsymbol{H}(\mathrm{div})$ spaces, *SINUM*, *2007*.

Tz. Kolev and P. Vassilevski, Parallel auxiliary space AMG for $\boldsymbol{H}(\mathbf{curl})$ problems, *JCM*, *2009*.

hypre, `http://www.llnl.gov/CASC/hypre`.